TITLE: FINE-TUNING TROCR FOR HANDWRITING RECOGNITION USING SYNTHETIC DATA

NAME: DEVANSH SENGAR

Role: AI Engineer

Platform: Google Colab (T4 GPU)

Model: microsoft/trocr-base-handwritten

1. Objective

This project goal was to fine-tune a transformer-based OCR model, TrOCR, to recognize handwritten text. The project focused on running a full training and evaluation pipeline on the IAM handwriting dataset and designing it to not exhaust the trainer's limited computing resources. Ultimately the goal was to obtain a low Character Error Rate (CER) and Word Error Rate (WER) and prove the model was functional for handwritten samples.

2. Dataset

We used the IAM Handwriting Dataset, a widely recognized benchmark for handwriting recognition. It contains scanned handwritten text with corresponding transcriptions. Due to the dataset's size and complexity, preprocessing and tokenization were carefully implemented to fit within Google Colab's resource limits. The dataset was split into training and test sets for fine-tuning and evaluation, respectively.

3. Model Choice & Justification

The chosen model was microsoft/trocr-base-handwritten, a Vision Transformer encoder combined with a Transformer decoder, optimized for handwriting OCR. This version was preferred over larger variants because it balances accuracy with computational feasibility on limited GPU memory (T4 with 16GB VRAM).

4. Preprocessing

- Input images were resized and normalized to 384×384 RGB format to match the model's expected input size.

- Labels were tokenized using the TrOCR processor, ensuring proper format for training.
- DataLoader was used to batch and shuffle the dataset efficiently, preventing memory overload.
- Additional preprocessing included error handling to avoid incompatible input dimensions.

5. Fine-Tuning Strategy

- Training was conducted for 10 epochs with mixed precision enabled to accelerate computation and reduce memory usage.
- AdamW optimizer was used with a learning rate of 5e-5.
- Batch size was set considering GPU constraints, and gradient accumulation was employed where necessary.
- Training progress was monitored using average loss per epoch.
- Post-training, the model was saved for evaluation.

6. Evaluation Metrics

- The evaluation was done on the test set using Character Error Rate (CER) and Word Error Rate (WER).
- The model achieved an average CER of 4.77% and WER of 11.55%, which meets the target accuracy for practical handwriting recognition applications.
- Evaluation was conducted using custom code to decode predictions and compare with ground truth, converted into percentages for clearer interpretation.

7. Challenges Faced

- Initial runtime crashes due to RAM exhaustion required optimizing batch size and data loading.
- Tokenizer and model configuration errors (such as missing decoder_start_token_id and pad_token_id) required debugging and setting proper configuration parameters.
- Managing the IAM dataset's complexity and size within Colab's memory limits was challenging, requiring careful preprocessing and efficient batching.

- Installing and managing external libraries for metrics calculation involved resolving dependency issues.

8. Improvements (Future Work)

- Experiment with data augmentation techniques such as noise addition, skewing, or occlusion to improve model robustness.
- Fine-tune with larger batch sizes on more powerful hardware for potentially better convergence.
- Incorporate additional datasets like Imgur5K to increase diversity and generalization.
- Explore hyperparameter tuning for learning rate, optimizer variants, and number of epochs to optimize performance further.
- Implement a more modular and documented codebase to improve maintainability and reproducibility.

9. Conclusion

The project showed that a transformer-based OCR model could be finetuned for handwritten text, and produced a competitive CER and WER within the limits and resources we had available. Even though resources and limited datasets presented challenges, the pipeline currently implemented is entirely functional, and can be potentially extended with more data and advanced methods. The performance shows that TrOCR-base can be a suitable model for handwriting recognition tasks, and with even more optimizations could be adapted to real-world document digitization applications.